UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/015,533	12/11/2001	Andrew J. Fish	10559-628001/P11062	4295

8791 7590 12/21/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
CHEN, TSE W	
ART UNIT	**PAPER NUMBER**
2116	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	12/21/2006	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/015,533
Filing Date: December 11, 2001
Appellant(s): FISH ET AL.

Joni D. Stutman-Horn
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed September 29, 2006 appealing from the Office action mailed March 1, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5835760	Harmer	11-1998
5978912	Rakavy et al.	11-1999

- Extensible Firmware Interface Specification -- Draft for Review, (January 19, 2000), Intel Corporation, Version 0.92, pp.1, 4, 9, 13, 104, 299.
- Kozierok, Charles M., "BIOS Updates and The Flash BIOS", (August 26, 2000), www.pcguide.com, pg. 1, http://web.archive.org/web/20000831062323/http://www.pcguide.com/ref/mbsys/bios/compFlash-c.html.
- Davis, Mark et al., "Draft Unicode Technical Report #10 - Unicode Collation Algorithm", (March 30, 1997), Revision 1.0, pg.2, http://www.unicode.org/unicode/reports/tr10/tr10-1.html.
- Newton's Telecom Dictionary, Telecom Books, Sixteenth Edition, (2000), pg. 614.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> [a] A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-3, 5-8 and 17-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Harmer, US patent 5,835,760 in view of Extensible Firmware Interface Specification – Draft for Review, hereinafter EFIS.

3. In re claim 1, Harmer discloses a system [200] comprising:

- Central processor [col.13, ll.40-41; associated with host].

- A non-volatile memory [rom] coupled with the central processor and storing platform firmware [bios] [col.8, ll.48-49].
- A machine-readable medium [mass memory storage; e.g., 114] coupled with the central processor, the machine-readable medium to be used in initializing the operating environment for the system upon power up [col.13, ll.45-47; expansion bios needed to run devices in operating environment], the machine-readable medium comprising a first set of instructions [128] [col.9, ll.49-54] forming at least a portion of the operating environment [col.9, ll.26-29; to run device], and a second set of instructions [120] [col.9, l.40] defining one or more firmware extensions [124] to enable the system to access the first set of instructions [124, a component of 120, accessed 128], wherein the one or more firmware extensions comprise a self-describing media module [col.9, ll.16-29; col.11, ll.34-36; system reads self describing 124 component to access other part of code such as 128 and 134 in order to run device].

4. Harmer did not disclose the details of an extensible firmware interface.

5. EFIS teaches an extensible firmware interface [EFI] comprising data tables having platform-related information [Page. 299, ll.3-7], a loader for an operating environment [Page 9, fig.1-1; Page 104, Section 4.4] and boot and runtime service calls available to the operating environment [Page 1, ll.3-4], wherein the EFI enables extension of platform firmware by loading driver and application images, which when loaded, have access to all EFI defined runtime and boot services [fig.2-1; Page 13, ll.1-3].

6. The motivation for incorporating the External Firmware Interface includes the benefit of abstraction, such that code may be written for a variety of hardware devices without having explicit knowledge of the specifics for each device [EFIS: Page 4, ll.3-5].

7. Accordingly, it would have been obvious to a person of ordinary skill in the art at the time of invention to incorporate EFI as taught by the EFIS with the system disclosed by Harmer for the benefit of permitting faster and easier development of code for a variety of hardware devices.

8. As to claim 2, Harmer discloses the machine-readable medium comprises a hard disk platter [col.13, ll.47-50].

9. As to claim 3, Harmer discloses the one or more firmware extensions comprise a file system driver to support a file system format not supported by the platform firmware [col.9, ll.49-54; 132 necessary to run device with 130].

10. As to claim 5, Harmer discloses the central processor comprises a central processing unit housed in a single chip [col.1, ll.31-35].

11. As to claim 6, Harmer discloses a volatile memory [ram] [col.13. l.41]; and a motherboard coupling the volatile memory, the non-volatile memory and the machine-readable medium with the central processing unit [fig.9; motherboard by definition connects the main components of a computer system; although not explicitly mentioned, it is considered inherent to the operation of the system].

12. As to claim 7, Harmer discloses self-describing machine-readable medium [114; col.9, ll.49-54; reads self describing 124 component to access other part of code such as 128 and 134 in order to run device] comprising:

- A first set of instructions [120] in a first portion of the medium [fig.5] [col.9, ll.49-54] defining operations for enabling a machine to access a second set of instructions [128] in a second portion of the medium [fig.5] [col.11, ll.34-36; 124, a component of 120, accesses 128] comprising at least a portion of an operating system [operating data] stored on the machine-readable medium in a format that is unreadable by the machine before the machine loads the first set of instructions [col.9, ll.49-54; reads self describing 124 component to access other part of code such as 128 and 134 in order to run device with 130].
- The second set of instructions [128] [fig.5].

13. Harmer does not disclose incorporating an extensible firmware interface.

14. EFIS teaches wherein the first set of instructions comprises at least one extensible firmware interface image [EFI] providing a software abstraction enabling access to the second portion of a medium, wherein platform firmware of the machine does not have a mechanism to access the second portion of the medium prior to accessing the EFI image [Page 1, ll.3-4; fig.2-1; Page 13, ll.1-3]

15. The motivation for incorporating the EFI includes the benefit of abstraction, such that code may be written for a variety of hardware devices without having explicit knowledge of the specifics for each device [Page 4, ll.3-5].

16. Accordingly, it would have been obvious to a person of ordinary skill in the art at the time of invention to incorporate an EFI as taught by the EFIS with the system disclosed by Harmer for the benefit of permitting faster and easier development of code for a variety of hardware devices.

17. As to claim 8, Harmer discloses the first set of instructions comprise one or more extensions to platform firmware capability [col.13, l.63 - col.14, l.2].

18. As to claim 17, Harmer and EFIS disclose each and every limitation of the claim involving the means thereof [machines readable medium relates to mass storage means] as discussed above in reference to claims 1 and 7.

19. In re claim 18, Harmer discloses the mass storage means comprises an optical disk [compact disk] [col.13, ll.47-50].

20. In re claim 19, Harmer discloses the means for extending platform firmware capabilities comprise a file system driver to support file system format not supported by the platform firmware [col.9, ll.49-54; reads self describing 124 component to access other part of code such as 128 and 134 in order to run device with 130].

21. Claims 4 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Harmer and EFIS as applied to claims 1 and 17 above, and further in view of BIOS Updates, hereinafter BIOSU.

22. Harmer and EFIS disclose each and every limitation as discussed above in reference to claims 1 and 17. Harmer did not disclose a non-volatile memory that comprises a random access non-volatile memory.

23. BIOSU teaches the non-volatile memory comprises random access non-volatile memory [eeprom] [Paragraph 2, ll.5-6].

24. The motivation for using a random access non-volatile memory, in this case an EEPROM, allows for "a ROM that can be erased and re-written" [BIOSU: Paragraph 3, l.3]. This will allow for updates to be made to the BIOS without requiring physical replacement of ROM.

25. Accordingly, it would have been obvious to a person of ordinary skill in the art to modify the device disclosed by Harmer to incorporate a non-volatile random access memory as described by BIOSU for the benefit of providing a circuit housing a BIOS that is more readably modifiable.

26. Claims 9-11 and 13-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Harmer and EFIS as applied to claims 7 and 8 above, and further in view of Rakavy et al., US Patent 5978912, hereinafter Rakavy.

27. In re claim 9, Harmer discloses the portion of an operating system comprises operating data that may include, but is not limited to, system configuration information, data, text, passwords, or any other information that may provide some purpose during the start-up of the system [col.16, ll.20-24]. Harmer did not disclose explicitly the operating data includes an operating system loader.

28. Rakavy teaches the POST reads a block of data from a predetermined location from the boot device, usually the hard disk or a diskette drive, into memory, and passes control to the program in that data block. This program, known as a bootstrap loader, then loads a larger program into memory. If the larger program is properly loaded into memory the boot program passes control to it. The operating system is then initialized and gains control of the CPU [col.2, ll.27-34].

29. Rakavy provides this as background for the methodology of the typical startup procedure of an IBM compatible personal computer" [col.1, ll.64-66].

30. This standard behavior would accordingly suggest that it would be obvious to a person of ordinary skill in the art that, though Harmer does not specifically mention an operating system or

bootstrap loader, his invention would follow this standard startup procedure and provide such a program because it serves an important purpose during the start-up of the system.

31. As to claim 10, Harmer discloses the one or more extensions to platform firmware capability comprise a file system driver to support a file system format used to store at least a portion of the second set of instructions [col.11, ll.34-36; the file system described consists of giving the first portion of the expansion BIOS the ability to find and read the second portion of the expansion BIOS].

32. As to claim 11, Harmer discloses the one or more extensions to platform firmware capability comprise glyphs that represent a language [col.15, ll.57-62; glyphs are graphical in nature].

33. In re claim 13, Harmer discloses a machine-implemented method for extending platform firmware capabilities [col.8, ll.41-44], the method comprising:

- Loading on a system one or more firmware extensions [col.8, ll.41-44] from a boot media [col.46-47].
- Booting the system [col.13, ll.53-56].
- Loading and running operating data [that] may include, but is not limited to, system configuration information, data, text, passwords, or any other information that may provide some purpose during the start-up of the system from the boot media [col.16, ll.20-24] using the one or more loaded firmware extensions [col.15, ll.49-53], the one of more loaded firmware extensions [124] enabling the system to access the operating data from a portion of the boot media inaccessible to the unextended platform firmware [col.9,

ll.49-54; 124 component necessary to access other part of code such as 128 and 134 in order to run device with 130].

34. Harmer did not disclose explicitly the loading and running an operating system loader from the boot media using the one or more loaded firmware extensions.

35. Rakavy teaches the POST reads a block of data from a predetermined location from the boot device, usually the hard disk or a diskette drive, into memory, and passes control to the program in that data block. This program, known as a bootstrap loader, then loads a larger program into memory. If the larger program is properly loaded into memory the boot program passes control to it. The operating system is then initialized and gains control of the CPU [col.2, ll.27-34].

36. Rakavy provides this as background for the methodology of the typical startup procedure of an IBM compatible personal computer [col.1, ll.64-66].

37. This standard behavior would accordingly suggest that it would be obvious to a person of ordinary skill in the art that, though Harmer does not specifically mention an operating system or bootstrap loader, his invention would follow this standard startup procedure and provide such a program because it serves an important purpose during the start-up of the system.

38. Harmer and Rakavy do not disclose firmware extensions being compatible with an extensible firmware interface.

39. EFIS teaches an extensible firmware interface [EFI] comprising data tables having platform-related information [Page 299, ll.3-7], a loader for an operating system [Page 9, fig.1-1; Page 104, Section 4.4] and boot and runtime service calls available to the operating system [Page 1, ll.3-4], wherein the EFI enables extension of platform firmware by loading driver and

application images, which when loaded, have access to all EFI defined runtime and boot services, the system having an EFI architecture [fig.2-1; Page 13, ll.1-3].

40. The motivation for incorporating the EFI includes the benefit of abstraction, such that code may be written for a variety of hardware devices without having explicit knowledge of the specifics for each device [EFIS: Page 4, ll.3-5]

41. Accordingly, it would have been obvious to a person of ordinary skill in the art at the time of invention to render the firmware extensions to be compatible with an EFI as taught by the EFIS with the system disclosed by Harmer for the benefit of permitting faster and easier development of code for a variety of hardware devices.

42. As to claim 14, Harmer discloses loading one or more firmware extensions from a boot media during a system boot in such a manner that abstracts a mass storage device containing the boot media [col.13, ll.47- 50]. Harmer does not disclose the method for this abstraction as incorporating a block input/output protocol. Rakavy further teaches POST reads a block of data from a predetermined location from the boot device, usually the hard disk or a diskette drive [col.2, ll.27-29].

43. As to claim 15, Harmer further discloses the one or more firmware extensions comprise a file system driver to support a file system format used to store data on the boot media [col.11, ll.34-36; the file system described consists of giving the first portion of the expansion BIOS the ability to find and read the second portion of the expansion BIOS].

44. As to claim 16, Harmer further discloses: the one or more firmware extensions further comprise glyphs that represent a language [col.15, ll.57-62, glyphs are graphical in nature].

45. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rakavy, Harmer and EFIS as applied to claim 9 above, and further in view of Unicode Technical Report #10, hereinafter UTR.

46. Harmer discloses the general concept of storing information required during the start-up of the system may include a variety of operating data, text, or other information that increases the functionality of the system during the start-up of the system [col.15, ll.54-57]. Harmer did not disclose the inclusion of a Unicode collation module as an extension to a system that may be stored on a mass memory storage device.

47. However, UTR shows a Unicode Collation Algorithm is a well-known method for providing alphabetic, diacritic and case ordering [Page 2; Section Summary; Paragraph 3, ll.4-6].

48. The motivation behind ordering/collation is that sorted entities are far more searchable than ones that are not.

49. Sorting is a fundamental task in computers and it would be obvious to a person of ordinary skill in the art to modify Harmer to incorporate a Unicode collation algorithm as a method of increasing the functionality of a computer system without increasing the cost of the peripheral device and/or the system [Harmer: col.15, ll.52-53].

(10) Response to Argument

A. Claims 1-3, 5-8 and 17-19 are unpatentable over Harmer in view of EFIS.

1. Harmer teaches that the machine-readable medium is a self-describing media module forming at least a portion of the operating environment for use in initializing the system upon power up.

Appellant argues that Harmer "fails to teach or suggest that the machine-readable medium is a self-describing media module forming at least a portion of the operating environment for use in initializing the system upon power up" in reference to the limitation:

> "a machine-readable medium coupled with the central processor, the machine-readable medium to be used in initializing the operating environment for the system upon power up, the machine-readable medium comprising a first set of instructions forming at least a portion of the operating environment, and a second set of instructions defining one or more firmware extensions to enable the system to access the first set of instructions, wherein the one or more firmware extensions comprise a self-describing media module".

Examiner disagrees and submits that Harmer discloses:

- A machine-readable medium [mass memory storage; e.g., 114] coupled with the central processor [col.13, ll.40-41; associated with host].
- The machine-readable medium to be used in initializing the operating environment for the system upon power up [col.13, ll.45-47; expansion bios needed to run devices in operating environment].
- The machine-readable medium comprising a first set of instructions [128] forming at least a portion of the operating environment [col.9, ll.26-29, ll.49-54; to run device].
- A second set of instructions [120] [col.9, l.40] defining one or more firmware extensions [124] to enable the system to access the first set of instructions [124, a component of 120, accesses 128].
- Wherein the one or more firmware extensions comprise a self-describing media module [col.9, ll.16-29; col.11, ll.34-36; system reads self describing 124 component to access other part of code such as 128 and 134 in order to run device].

Thus, Harmer clearly discloses ***the machine-readable medium*** [mass memory storage; e.g., 114] ***that is a self-describing media module*** [col.9, ll.16-29; col.11, ll.34-36; system reads self

describing 124 component to access other part of code such as portion 128 in order to run device] ***forming at least a portion of the operating environment for use in initializing the system upon power up*** [col.9, ll.26-29; col.13, ll.45-47; portions 124, 120, 128 part of operating environment that initializes and *operates*].

Appellant further argues that "Harmer does not teach a firmware extension that is related to the operating environment..." and admits that Harmer "teaches a firmware extension that is meant to be loaded on the host computer 'to properly initialize and operate device 114'". Examiner points to Appellant's admission and submits that a firmware that initializes and operates a device can be considered part of an operating environment.

Applicant further argues that "claim 1 requires that the medium comprises ***a second set of instructions defining one or more firmware extension to enable the system to access the first set of instructions***... as explicitly recited in the claim, it is the second set of instructions that form the 'device driver', or file system driver information". Examiner was not able to find where in claim 1 exists the explicit reciting that the second set of instructions actually forms the ***device driver*** or file system driver information, as it appears Appellant has read limitations [e.g., device driver and associated information] from the specification into the claim. Examiner submits that Harmer clearly discloses ***a second set of instructions*** [first portion 120] ***defining one or more firmware extensions*** [initialization code 124] ***to enable the system to access the first set of instructions*** [second portion 128] [col.9, ll.16-29, ll.49-54; col.11, ll.34-36; system reads 124 of portion 120 to access another portion 128 in order to run device]. Thus, Appellant's arguments relating the claim limitations to a device driver or file system driver information is unfounded.

Appellant attempts to establish that an operating environment is the same as an operating system by stating "a portion of the operating environment, i.e., a portion of the operating system". Examiner disagrees as operating environments are considered to be entities/systems that *encompass* operating systems [Newton's Telecom Dictionary, pg.614 where ***two separate distinct*** entries exist for operating environment and operating system]. Furthermore, Examiner was not able to find anywhere in the original specification where an operating environment is defined explicitly as an operating system. Thus, Appellant's arguments relating the claim limitations to an operating system [e.g., DOS] instead of an operating environment [i.e., system that includes an operating system and other applications such as the software portions of Harmer] is unfounded.

Appellant argues that Harmer "teaches only one set of instructions..." Examiner disagrees as Harmer clearly discloses more than one set of instructions [col.9, ll.16-29, ll.49-54; col.11, ll.34-36; system reads one set of instruction 124 of instruction set 120 to access another set of instruction 128 in order to run device].

Appellant argues that Harmer "fails to teach or suggest a self-describing media device that comprises both instructions to operate the device and instructions forming a portion of the operating environment, as recited in claim 1". Examiner disagrees and points to Appellant's admission that Harmer "teaches a firmware extension that is meant to be loaded on the host computer 'to properly initialize and operate device 114'" as signifying instructions that operate a device and considered to be part of an operating environment.

Appellant argues that Harmer "does not teach that the medium is a boot media... that the firmware extension resides on boot media to enable reading and loading of a portion of the

operating environment – as inherent in the definition of boot media". Examiner disagrees and submits that Harmer is associated with a boot media as exemplified by figure 12B that clearly illustrates the startup [boot] process with a medium [mass memory storage device] in an operating environment. Additionally, Examiner submits that claim 7 does not explicitly recite "boot media", as it appears Appellant has read limitations [e.g., boot] from the specification into the claim. Thus, Appellant's arguments relating the claim limitations to a "boot media" is unfounded.

Appellant argues that EFIS does not teach "that the boot media containing an OS loader may be on mass storage that is unreadable... claim 7 requires that 'the machine readable medium in a format that is unreadable by the machine before the machine loads the first set of instructions". Examiner submits that claim 7 was rejected based on a combination of Harmer in view of EFIS. Harmer, the primary reference, was used to teach "***a portion of an operating system stored on the machine readable medium in a format that is unreadable by the machine before the machine loads the first set of instructions***" [col.9; expansion BIOS portion 128 not readable until portion 124 of 120 is read first]. Appellant's argument concerning the bodily incorporation of secondary reference EFIS's teaching of OS loader into the structure of primary reference Harmer is not the test for obviousness – rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. Clearly, it would have been obvious to one of ordinary skill in the art to incorporate the EFI teachings of EFIS to Harmer [i.e., EFI as *first set of instructions* allows EFI's abstraction to access *unknown second portion* such as an *operating system loader*] as EFI's abstraction permits faster and easier development of code for a variety of hardware devices [EFIS: pg.4, ll.3-5].

Appellant argues about claim 17 as above. As such, Examiner maintains positions against Appellant's arguments as discussed above.

2. There is motivation to combine EFIS with Harmer.

Appellant argues that there "is no motivation to combine EFIS with Harmer". Examiner disagrees as explicit motivation was cited as being the "abstraction that permits code to be written for a variety of hardware devices without having explicit knowledge of the specifics for each device", permitting faster and easier development of code for a variety of hardware devices [EFIS: pg.4, ll.3-5]. Examiner submits that writing specific code for different devices can be costly and inefficient. EFIS's teaching offers an abstraction that permits a developer to essentially write one universal code that can be used by many different devices. In other words, writing one set of code is more efficient than writing many sets of different codes. Thus, one with ordinary skill in the art would be motivated to incorporate EFIS's teachings to provide efficient development of code for a variety of hardware devices.

Appellant's recitation of the various court opinions is not dispositive of patentability as patentability is determined on a case-by-case basis. In the present case, Examiner submits the following:

- Primary reference Harmer discloses each and every limitation of the claimed system, except for using the EFI standard.
- Secondary reference EFIS discloses the EFI standard.
- EFI standard offers abstraction, allowing developers to efficiently develop one universal set of code instead of many sets of different codes for a variety of different devices.

Thus, it would have been obvious to one of ordinary skill in the art, having the teachings of Harmer and EFIS before him at the time the invention was made, to incorporate the EFI standard taught by EFIS to Harmer's system, in order to obtain an EFI standard system. One of ordinary skill in the art would have been motivated to make such a combination as it provides an efficient way to write one set of code for a variety of hardware devices without having explicit knowledge of the specifics for each device.

B. Claims 4 and 20 are unpatentable over Harmer and EFIS in view of BIOS Updates.

Appellant argues with respect to Harmer and EFIS as above. As such, Examiner maintains positions against Appellant's arguments as discussed above.

C. Claims 9-11 and 13-16 are unpatentable over Harmer and EFIS in view of Rakavy et al.

Appellant's argument concerning the bodily incorporation of Rakavy's teaching of OS loader into the structures of Harmer and EFIS is not the test for obviousness – rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. Examiner submits that Harmer and EFIS disclose each and every limitation of the claimed system, except for the well-known operating system loader [i.e., Harmer discloses operating data to be anything that may "provide some purpose during the start-up of the system" (col.16, ll.20-24), but did not explicitly stipulate the operating data to include an operating system loader]. Even though the operating system loader should be considered well known as "providing some purpose during the start-up of the system", Examiner still provided explicit factual reference with Rakavy that discloses an operating system loader in ***a typical startup procedure*** [col.1, ll.64-66]. Clearly, it would have been obvious to one of ordinary skill in the art

to incorporate the explicit operating system loader teachings of Rakavy to Harmer and EFIS [e.g., explicitly as part of the operating data that "provides some purpose during the start-up of the system"] as operating system loaders are well known in the art and provides an important purpose [i.e., loading the operating system] during the start-up of the system.

D. Claim 12 is unpatentable over Rakavy et al., Harmer and EFIS in view of UTR.

Appellant argues with respect to Rakavy et al., Harmer and EFIS as above. As such, Examiner maintains positions against Appellant's arguments as discussed above.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Tse Chen

November 22, 2006

Conferees:

Lynne Browne

Lynne H. Browne
Appeal Specialist, TQAS
Technology Center 2100

Rehana Perveen

REHANA PERVEEN
SUPERVISORY PATENT EXAMINER
12/19/06